IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Gabriel Aaron Cohen et al | Examiner: Nicholas Augustine

Serial No.:   10/727,275 | Art Unit   2179

Filing Date: 12/03/2003

Title: Self-configuring component for

              recognizing and transforming

              host data

Mail Stop Appeal Brief-Patents

Commissioner for Patents

P. O. Box 1450

Alexandria, VA 22313-1450

BRIEF ON APPEAL

This Brief supports the appeal to the Board of Patent Appeals and

Interferences from the final rejection dated 03/19/2009, in the application

identified above. Appellant filed the Notice of Appeal on 5/01/2009, and

now submits this Brief as required by 37 C.F.R. §1.192(a). This Brief

addresses the issues raised in the final rejection.

TABLE OF CONTENTS

I.   REAL PARTY IN INTEREST

The real party in interest in this appeal is International Business Machines Corporation pursuant to the Assignment recorded on 12/03/2003, at Reel 014767 and Frame 0311.

II.   RELATED APPEALS AND INTERFERENCES

With respect to the appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in this appeal, to the best of the knowledge of the undersigned, there are none.

III.   STATUS OF CLAIMS

Claims 1-21 were finally rejected in an Office Action dated 03/19/2009. Claims 1-21 are the pending claims that are the subject of this appeal and are set forth in the attached Claims Appendix.

IV. STATUS OF AMENDMENTS

All amendments have been entered into the record. No amendment has been filed subsequent to final rejection.

V.   SUMMARY OF CLAIMED SUBJECT MATTER

Claim 1 relates to a computer-readable medium having stored computer program code that when executed by a computer causes a computer system to recognize a character-based user interface having a plurality of host components and to transform the character-based user interface to a web enabled user interface. A screen image for the computer user interface is scanned by a plurality of software agents. Each agent is designed to determine the existence of a different host component user interface type unique to the agent. An agent defines a match region for each host component type that the agent finds in the character-based user interface (in the application, see line 12 of paragraph 24, page 9 to line 4 of page 10). The agent also determines whether two or more match regions overlap (see Fig. 4B and paragraph 27. Also see Fig. 7, steps 740 and 750). The match regions associated with each agent are then rendered to compose the web enabled user interface (see lines 1-3 of paragraph 37, page 15. Also see Fig. 7, step 760).

Claim 2, dependent from claim 1, further recites that the rendering code renders each match region as a widget and that the aggregation of widgets form a formatted page (see paragraph 35, page 14, lines 5-8).

Claim 3, dependent from claim 1, adds that the computer-readable medium further contains code for resolving a conflict between overlapping regions based on a policy to determine which agent associated with a match region controls the overlap region (see paragraph 28).

Claim 4 further refines the resolving code of claim 3 by assigning a predetermined priority to each agent. Claim 4 further recites code for comparing the predetermined priority of the two or more conflicting agents; and code for selecting the agent with the highest predetermined priority to control the overlapping region (see paragraph 28).

Claim 5 states that the resolving code of claim 3 compares the size of conflicting regions which overlap; and further contains code for selecting the agent having the smaller size region to control the overlapped region (see paragraph 29).

Claim 6 states that the resolving code of claim 3 further comprises code for assigning a dynamic priority to each conflicting region, the dynamic priority being based on the projected amount of time expended to render each conflicting region; and further code for selecting the agent controlling the conflicting region having the highest priority to retain control over the overlapping region (see paragraph 30).

Claim 7, dependent from claim 4, states that the computer-readable

medium of claim 4 further contains code for controlling the conflicting agents to negotiate whether to relinquish control of an overlap region (see paragraph 32).

Independent claim 8 recites a computer system using a plurality of agents to scan a character-based interface, where each agent is designed to detect a single character pattern in the character-based user interface (see line 12 of paragraph 24, page 9 , to line 4 of page 10).

Claim 9, dependent from claim 8, requires that each agent render its output as a widget, wherein the aggregation of widgets forms an output page.

Claim 10, dependent from claim 8, requires an agent manager for determining whether two or more match regions overlap.

Claim 11 requires the resolution of a conflict between overlapping match regions based on a policy to determine which agent associated with one match region controls the overlap region.

Claim 12 requires assigning a predetermined priority to each agent to control the overlapping regions.

Claim 13, dependent from claim 11 defines the policy for resolving overlapping match regions as comparing the size of the conflicting regions which overlap; and selecting the agent having the smaller size region to

control the overlapped region.

Claim 14, dependent from 11, defines the policy for resolving overlapping match regions as assigning a dynamic priority to each conflicting based on the projected amount of time expended to render each conflicting region.

Independent claim 15 and its dependent claims 16 through 21 claim a method for transforming a character-based user interface to a web enabled user interface in language substantially equivalent to that of claims 1 through 7, but as a method (see line 12 of paragraph 24, page 9 , to line 4 of page 10; Fig. 4B and paragraph 27. Also see Fig. 7, steps 740 and 750; see lines 1-3 of paragraph 37, page 15. Also see Fig. 7, step 760).

VI.   GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Whether claims 1 through 7 and claims 15 through 20 are anticipated under 35 U.S.C. §102(b) by U.S. patent publication 2002/0091818, hereinafter "Cascio".

Whether claims 8 through 14 are rendered obvious under 35 U.S.C. 103 by the combination of Cascio in view of US Publication 2002/0120725, hereinafter "DeCosta".

VII.   ARGUMENT

Rejection under 35 U.S.C. §102(b)

Anticipation requires the disclosure in a single prior art reference of each element of the claim under consideration. See W. L. Gore & Assocs. v. Garlock, Inc. 721 F.2d 1540, 220 USPQ 303, 313 (Fed. Cir. 1983). .

In the final Office Action dated 3/19/2009, the Examiner rejected claims 1-7 and 15-21 under 35 U.S.C. §102(b) as being unpatentable over Cascio. Claims 2-7 ultimately depend from claim 1, and claims 16-21 ultimately depend from claim 15. Claims 1 through 7 and claims 15 through 21 stand or fall together. Claims 8 through 14 do not stand or fall with claims 1 through 7 and claims 15 through 21, as separately argued below. As will be shown below, Cascio does not disclose, teach or suggest all elements of each claim rejected under 35 U.S.C 102(b).

Claims 1 through 7 recite a computer storage medium containing computer code and claims 15 through 21 recite a method_of transforming a character-based user interface into a web based interface, including the capability of detecting regions of overlapping character-based data and resolving the overlapping regions. The Examiner and applicants disagree about whether or not Cascio teaches such a capability. Applicants assert that Cascio does not teach such a capability, and that, in fact, Cascio's

disclosed embodiment is incapable of performing such a capability.

In paragraph 4 of the final Office Action of 3/19/2009, the Examiner

states that:

> "As claim 1, Cascio teaches a computer-readable medium whose contents cause a computer system (fig. 1, label 30; par [0044]) to recognize a character based user interface having a plurality of host component types and to transform the character based user interface to a web enabled user interface (Abstract; par [0026]; par [0047]"

Examiner further states in paragraph 4 that Cascio teaches:

> "the computer system having instructions to perform the steps of (par [0044]; [par 0047]): code for scanning the character based user interface by a plurality of agents (par[0047]; par[0055]; par [0075] ... "

Applicants assert that Cascio does not disclose the use of plural

agents as required by the independent claims 1 and 15 in issue. Fig. 6 of

Cascio reveals that Cascio discloses a single computer process that

sequentially compares a series of rules to user interface data until the data

is found to match all components of a single rule.   Paragraph [0047] of

Cascio begins by stating that " In the preferred embodiment, the present

invention is implemented as one or more computer software programs."

However, there is no further description of what this might mean or what

functions that plural software programs might individually perform. The remainder of [0047] contains statements of a general nature believed to be not relevant to the present claims.

Paragraph [0055] of Cascio states that user interface data patterns can be specified by the use of regular expression syntax. Regular expressions are rules that typically use wild-card nomenclature to define search patterns in textual data. However, the use of general expressions is believed not relevant to any of the claims in prosecution. No claim in prosecution asserts the use of regular expressions.

Paragraph [0075] generally describes Fig. 10 of Cascio. Paragraph [0036] states that Fig. 10 depicts an example of a markup language output document that corresponds to an extracted portion of a host screen. While the present disclosure teaches the use of a markup language to support the transformation to a web-based environment, the rejected claims do not contain any such explicit limitation.

Examiner further states in paragraph 4 of the final Office action with regard to the Cascio specification:

> "determining whether two or more match regions overlap (fig. 6; par [0058]; par [0072]-[0073]; par [0076]-[0077], that by comparing complex data components (windows or objects) against rules will determine if regions/areas overlap"

Cascio paragraph [0058] describes the identification of a matching text region using both textual characters and attributes of textual characters. Attributes of character-based user interfaces include such things as font color, or font size, and so forth. However, again, no claim in prosecution requires the use of attributes. Therefore, paragraph [0058] does not appear to be relevant to any of the claims in issue.

Cascio paragraphs [0072]-[0073] describe the steps shown in the flowchart of Fig. 6. Fig. 6 shows a sequential process of applying data pattern rules to input character data (see steps 610 through 650) until a rule is found that matches the data. The matching data is then extracted (see steps 650 and 660). The corresponding description in paragraphs [0072]-[0073] does not expand the process shown in Fig. 6 and, in particular, do not describe, teach or suggest the detection of overlapping character-based regions and the resolution of such overlapping regions.

Cascio paragraphs [0076]-[0077] contain summation statements that also do not disclose, teach or suggest the detection of overlapping character-based regions and the resolution of such overlapping regions.

Independent claims 1 and 15 of the present application require detection of overlapping character-based regions. For example, claim 1, lines 9-13 contain the following limitations:

code for defining a match region for each host component type

found to exist by an agent in the character-based user interface;

code for determining whether two or more match regions

overlap;

and code for rendering match regions associated with each

agent to compose the web.

Claim 3, dependent from claim 1, requires "code for resolving a

conflict between two or more match regions which overlap...." Examiner

cites the same paragraphs [0052], [0072]-[0073] and [0076]-[0077]

discussed above against claim 3, none of which disclose or teach detection

and resolution of overlapping regions.

Claim 4 narrows the resolving code by assigning a predetermined

priority to each agent. The Examiner adds paragraph [0065] to the

paragraphs already discussed for the proposition that Cascio assigns

predetermined priority to overlapping regions, Paragraph [0065] however,

merely states that simple text editors can be used to form text and attribute

patterns.

Claim 5 uses the size of conflicting regions to resolve the overlap.

The Examiner cites Cascio Fig. 9, labels 920 and 930 against this claim.

Fig. 9, however, merely shows an example GUI screen, in which no

regions overlap. Labels 920 and 930 are two separate regions of different size.

Claim 6 assigns priorities to overlapping regions in a dynamic manner. The Examiner cites all of the above discussed Cascio paragraphs against this claim, none of which discuss overlapping regions and resolution priorities.

Claim 7 allows negotiation between agents to resolve an overlap. Cascio paragraphs [0052] and [0072]-[0073] are cited against this claim, neither of which discuss overlapping regions and resolution priorities.

Cascio does not disclose, teach or suggest the subject matter of any of claims 1 through 7. Claims 15 through 21 contain substantially equivalent language in a method format; therefore, these claims are patently distinct over Cascio as well.

Applicants further allege that Cascio is incapable of detecting overlapping regions by reason of the disclosed embodiment. As discussed above, Cascio's process shown in Fig. 6 and described in paragraphs [0072]-[0073] is a serial process that proceeds until all components of a rule are matched or rejected. Such a serial process retains no memory of past rule component comparisons other than the fact that the comparisons matched a rule component. Fig. 6 and its description alone confirm that

Cascio has not contemplated the detection of overlapping character-based regions, as recited in the present claims.

Support for the detection and resolution of overlapping regions is found in the application at:

Figs. 4A, 4B and 4C and paragraphs [0026] through [0032].

Rejection under 35 U.S.C. §103

Claim 8 and its dependent claims 9 through 14 are rejected as being obvious over Cascio in view of DeCosta (U.S. publication 2002/0120725). Claim 8 as originally filed contained the phrase "having an anti-virus program" in its preamble. DeCosta was cited by Examiner as disclosing an anti-virus program. The phrase "having an anti-virus program" was deleted in the first response of July 7, 2007 to the first Office action of April 12, 2007. Claim 8, as presently written, requires a plurality of agents, each agent for scanning a character-based user interface to determine a different character-based pattern. As described early in this Brief, Cascio does not disclose, teach or suggest a plurality of separate agents, each capable of detecting a different textual pattern. Cascio discloses only a serial process of pattern rule matching. Further, dependent claims 10 through 14 recite, respectively, overlapping regions (claim 10), resolution of overlapping regions by policy (claim 11), by predetermined priority (claim 12), by size of conflicting regions (claim 13) and by assigning a dynamic priority (claim 14). It has been

demonstrated that Cascio does not disclose or suggest any of the subject matter of

dependent claims 9 through 14. Therefore, these dependent claims patentably

distinguish over Cascio in their own right.

Conclusion

In conclusion, for the reasons discussed above , Appellants

respectfully request reversal of the Examiner's rejection of claims 1-7 and

15-21 under 35 USC 102(b) as being unpatentable over Cascio. Appellants

further respectfully request reversal of the Examiner's rejection of claims

8-14 under U.S.C. §103(a) as being unpatentable over Cascio in view of

DeCosta.

If there are any other fees due in connection with the filing of this

Brief on Appeal, please charge the fees to the Deposit Account 09-0461. If

a fee is required for an extension of time under 37 C.F.R. §1.136 not

accounted for above, such an extension is requested and should also be

charged to our Deposit Account.

Respectfully submitted,

Date: July 1, 2009   /Jerry W Herndon/
_____   _____
                                       Attorney of Record
                                       Reg. No. 27,901
                                       17 Crosswinds Estates
                                       Drive
                                       Pittsboro, NC 27312
                                       (919) 542-0345

VIII. CLAIMS APPENDIX

The claims on appeal are as follows:

    1. (rejected) A computer-readable medium having stored computer program code that when executed by a computer cause a computer system to recognize a character-based user interface having a plurality of host component types and to transform the character-based user interface to a web enabled user interface, the medium having code to control the computer, the medium containing:

        code for scanning the character-based user interface for a plurality of agents; code in each_agent for determining the existence of a different host component type unique to the agent;

        code for defining a match region for each host component type found to exist by an agent in the character-based user interface;

        code for determining whether two or more match regions overlap; and

        code for rendering match regions associated with each agent to compose the web enabled user interface.

    2. (rejected) The computer-readable medium of claim 1 wherein the rendering code further comprises:

        code rendering each match region as a widget, the aggregated

widgets composing a formatted output page.

3. (rejected) The computer-readable medium of claim 1 further comprising resolving code executed before the rendering code, comprising:

code for resolving a conflict between two or more match regions which overlap based on a policy to determine which agent associated with a match region controls the overlap region.

4. (rejected) The computer-readable medium of claim 3 wherein the resolving code comprises:

code for assigning a predetermined priority to each agent;

code for comparing the predetermined priority of the two or more conflicting agents; and

code for selecting the agent with the highest predetermined priority to control the overlapping region.

5. (rejected) The computer-readable medium of claim 3 wherein the resolving code further comprises:

code for comparing the size of the conflicting regions which overlap; and code for selecting the agent having the smaller size region to control the overlapped region.

6. (rejected) The computer-readable medium of claim 3 wherein the resolving code further comprises:

code for assigning a dynamic priority to each conflicting region having a common overlapping region, the dynamic priority based on the projected amount of time expended to render each conflicting region; and

code for selecting the agent controlling the conflicting region having the highest priority to retain control over the overlapping region.

7. (rejected) The computer-readable medium of claim 4 further containing code for controlling the conflicting agents to negotiate whether to relinquish control of at least the overlap region.

8. (rejected) A computer system for recognizing a character-based user interface having a plurality of host component types and transforming the character-based user interface to a web enabled user interface, the computer system comprising:

a memory comprising a plurality of agent objects to scan the character-based user interface, each agent object determining the existence of a different host component type from the other agents, each agent object defining a match region for each host component type found to exist in the character-based user interface, each agent object rendering itsassociated match region to compose the web enabled user interface; and

a processor for running the plurality of agent objects.

9. (rejected) The computer system of claim 8 wherein each agent

renders each match region as a widget, the aggregated widgets composing a formatted output page.

10. (rejected) The computer system of claim 8 wherein the memory further comprises:

an agent manager for determining whether two or more match regions overlap.

11. (rejected) The computer system of claim 10 wherein the system resolves a conflict between two or more overlapping match regions based on a policy to determine which agent associated with one match region controls the overlap region, the processor running the policy.

12. (rejected) The computer system of claim 11 wherein the policy executed by the processor comprises:

assigning a predetermined priority to each agent;

comparing the predetermined priority of the two or more conflicting agents; and selecting the agent with the highest predetermined priority to control the overlapping region.

13. (rejected) The computer system of claim 11 wherein the policy executed by the processor comprises:

comparing the size of the conflicting regions which overlap; and

selecting the agent having the smaller size region to control the

overlapped region.

14. (rejected) The computer system of claim 11 wherein the policy executed by the processor comprises:

assigning a dynamic priority to each conflicting region having a common overlapping region, the dynamic priority based on the projected amount of time expended to render each conflicting region; and

selecting the agent controlling the conflicting region having the highest priority to retain control over the overlapping region.

15. (rejected) A method for recognizing a character-based user interface having a plurality of host component types and transforming the character-based user interface to a web enabled user interface, the method comprising:

scanning the character-based user interface for a plurality of agents;

determining which host component types exist in the character-based user interface, each agent determining the existence of a different host component type from the other agents;

defining a match region for each host component type found to exist by an agent in the character-based user interface;

determining whether two or more match regions overlap;

and

rendering match regions associated with each agent to compose the web enabled user interface.

16. (rejected) The method of claim 15 wherein the render step further comprises the step of:

rendering each match region as a widget, the aggregated widgets composing a formatted output page.

17. (rejected) The method of claim 15 further comprising a step before the rendering step, the step comprising:

resolving a conflict between two or more match regions which overlap based on a policy to determine which agent associated with a match region controls the overlap region.

18. (rejected) The method of claim 17 wherein the policy comprises the steps of:

assigning a predetermined priority to each agent;

comparing the predetermined priority of the two or more conflicting agents; and selecting the agent with the highest predetermined priority to control the overlapping region.

19. (rejected) The method of claim 17 wherein the policy comprises the steps of: comparing the size of the conflicting regions which overlap; and

selecting the agent having the smaller size region to control the overlapped region.

20. (rejected) The method of claim 17 wherein the policy comprises the steps of:

assigning a dynamic priority to each conflicting region having a common overlapping region, the dynamic priority based on the projected amount of time expended to render each conflicting region; and

selecting the agent controlling the conflicting region having the highest priority to retain control over the overlapping region.

21. (rejected) The method of claim 18 wherein the conflicting agents negotiate whether to relinquish control of at least the overlap region.

IX.   EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.